* **How to Use**

1. Open the app.js file in visual studio and start the node server using “node app.js”
2. Go to the HTML file and then start the live server.
3. We have made a 7x7 Grid of the city so input any values in current location and final location from (0,0) to (6,6).
4. As soon as you hit get route. The webpage is reloaded and the reloaded page shows the directions in it the ambulance has to follow.
5. Copy the directions showed on the screen and then paste it in the format-changer.py so that it converts it into the format that can be used in micro-python for ESP32.

* **For Various Test Cases**

1. For Various test cases you can open find\_route.py and find **City Grid**.

(City Grid is made similar to image of map provided in this git repo).

1. You’ll see a 7x7 grid which represents every junction.
2. 0 means the junction is working and functional irrespective of the traffic at the junction and 1 means the junction is not functional.
3. Input any location in the HTML webpage and you’ll get the directions and you can check that with the city grid too.
4. Change the values of City Grid from 1 and 0 then you’ll create a new city grid with functional and dysfunctional junctions and the code will find the shortest and path and will provide you the directions to the destination that can be then pushed into traffic lights LED.

* **For hardware Emulation**

[esp32-micropython-ssd1306 - Wokwi ESP32, STM32, Arduino Simulator](https://wokwi.com/projects/305568836183130690)

* Go to the above provided link (as it is the only one that provides the free use of ESP32)
* Copy the Micro Python Code from the Original Code folder and paste it into the IDE provided beside the Hardware emulator.
* Hit Enter after changing it to desired values for directions.
* Click on Run and the directions will be pushed into the display.

**Limitations**

1. All the Dynamic Map api are paid and the using the opensource maps for this tasks provides us with a static map which breaks the code and stops it.
2. Hardware not easily available to emulate online and these platform does not provide the esp32 to create a webhook connection to the server and then python program would have directly sent the directions to web-server and then esp32 receives it from the server and would have directly pushed it into the LED screens